# Cardinality of Equivalence Relations 

Lukas Bulwahn

September 13, 2023


#### Abstract

This entry provides formulae for counting the number of equivalence relations and partial equivalence relations over a finite carrier set with given cardinality.

To count the number of equivalence relations, we provide bijections between equivalence relations and set partitions [4], and then transfer the main results of the two AFP entries, Cardinality of Set Partitions [1] and Spivey's Generalized Recurrence for Bell Numbers [2], to theorems on equivalence relations. To count the number of partial equivalence relations, we observe that counting partial equivalence relations over a set $A$ is equivalent to counting all equivalence relations over all subsets of the set $A$. From this observation and the results on equivalence relations, we show that the cardinality of partial equivalence relations over a finite set of cardinality $n$ is equal to the $n+1$-th Bell number [3].


## Contents

## 1 Cardinality of Equivalence Relations

1.1 Bijection between Equivalence Relations and Set Partitions ..... 2
1.1.1 Possibly Interesting Theorem for HOL.Equiv-Relations ..... 2
1.1.2 Dedicated Facts for Bijection Proof ..... 2
1.1.3 Bijection Proof ..... 3
1.2 Finiteness of Equivalence Relations ..... 4
1.3 Cardinality of Equivalence Relations ..... 4
2 Cardinality of Partial Equivalence Relations ..... 5
2.1 Definition of Partial Equivalence Relation ..... 5
2.2 Construction of all Partial Equivalence Relations for a Given Set ..... 6
2.3 Injectivity of the Set Construction ..... 7
2.4 Cardinality Theorem of Partial Equivalence Relations ..... 8

## 1 Cardinality of Equivalence Relations

theory Card－Equiv－Relations<br>imports<br>Card－Partitions．Card－Partitions<br>Bell－Numbers－Spivey．Bell－Numbers<br>begin

## 1．1 Bijection between Equivalence Relations and Set Parti－ tions

## 1．1．1 Possibly Interesting Theorem for HOL．Equiv－Relations

This theorem was historically useful in this theory，but is now after some proof refactoring not needed here anymore．Possibly it is an interesting fact about equivalence relations，though．

```
lemma equiv-quotient-eq-quotient-on-UNIV:
    assumes equiv \(A R\)
    shows \(A / / R=(U N I V / / R)-\{\{ \}\}\)
proof
    show \(U N I V ~ / / R-\{\{ \}\} \subseteq A / / R\)
    proof
        fix \(X\)
        assume \(X \in U N I V / / R-\{\{ \}\}\)
        from this obtain \(x\) where \(X=R "\{x\}\) and \(X \neq\{ \}\)
            by (auto elim!: quotientE)
        from this have \(x \in A\)
            using <equiv \(A\) 〉 equiv-class-eq-iff by fastforce
    from this \(\langle X=R\) " \(\{x\}\rangle\) show \(X \in A / / R\)
            by (auto intro!: quotientI)
    qed
next
    show \(A / / R \subseteq U N I V / / R-\{\{ \}\}\)
    proof
        fix \(X\)
        assume \(X \in A / / R\)
        from this have \(X \neq\{ \}\)
            using 〈equiv \(A R\) 〉in-quotient-imp-non-empty by auto
        moreover from \(\langle X \in A / / R\rangle\) have \(X \in U N I V / / R\)
            by (metis UNIV-I assms proj-Eps proj-preserves)
        ultimately show \(X \in U N I V / / R-\{\{ \}\}\) by simp
    qed
qed
```


## 1．1．2 Dedicated Facts for Bijection Proof

lemma equiv－relation－of－partition－of：
assumes equiv $A R$
shows $\{(x, y) . \exists X \in A / / R . x \in X \wedge y \in X\}=R$

```
proof
    show {(x,y). \existsX\inA // R. x\inX\wedge y\inX}\subseteqR
    proof
        fix xy
        assume xy\in{(x,y). \existsX\inA // R.x\inX\wedge y\inX}
        from this obtain x y and X where xy = (x,y)
            and}X\inA//R\mathrm{ and }x\inXy\in
            by auto
        from }\langleX\inA// R\rangle\mathrm{ obtain z where X=R" {z}
            by (auto elim: quotientE)
        show }xy\in
            using <xy = (x,y)\rangle\langleX=R" "{z}\rangle\langlex\in X\rangle\langley\inX\rangle\langleequiv A R>
            by (simp add: equiv-class-eq-iff)
    qed
next
    show }R\subseteq{(x,y).\existsX\inA // R. x\inX\wedge y\inX
    proof
        fix xy
        assume xy\inR
        obtain x y where xy=(x,y) by fastforce
        from }\langlexy\inR\rangle\mathrm{ have (x,y) GR
            using }\langlexy=(x,y)\rangle\mathrm{ by simp
        have R" {x}\inA // R
            using <equiv A R>< (x,y) \inR`
            by (simp add: equiv-class-eq-iff quotientI)
    moreover have x\inR" {x}
                using }\langle(x,y)\inR\rangle\langleequiv A R
                by (meson equiv-class-eq-iff equiv-class-self)
    moreover have y\inR" {x}
                using }\langle(x,y)\inR\rangle\langleequiv A R> by sim
    ultimately have (x,y)\in{(x,y).\existsX\inA // R. x\inX\wedge y\inX}
                by auto
    from this show xy \in{(x,y).\existsX\inA // R. x \in X\wedge y \inX}
            using <xy = (x,y)> by simp
    qed
qed
```


### 1.1.3 Bijection Proof

lemma bij-betw-partition-of:
bij-betw $(\lambda R$. $A / / R)\{R$. equiv $A R\}\{P$. partition-on $A P\}$
proof (rule bij-betw-byWitness[where $\left.\left.f^{\prime}=\lambda P .\{(x, y) . \exists X \in P . x \in X \wedge y \in X\}\right]\right)$
show $\forall R \in\{R$. equiv $A R\} .\{(x, y) . \exists X \in A / / R . x \in X \wedge y \in X\}=R$
by (simp add: equiv-relation-of-partition-of)
show $\forall P \in\{P$. partition-on $A P\} . A / /\{(x, y) . \exists X \in P . x \in X \wedge y \in X\}=P$
by (simp add: partition-on-eq-quotient)
show $(\lambda R . A / / R)$ ' $\{R$. equiv $A R\} \subseteq\{P$. partition-on $A P\}$
using partition-on-quotient by auto
show $(\lambda P .\{(x, y) . \exists X \in P . x \in X \wedge y \in X\})$ ' $\{P$. partition-on $A P\} \subseteq\{R$.
equiv $A R\}$
using equiv-partition-on by auto
qed
lemma bij-betw-partition-of-equiv-with-k-classes:
bij-betw $(\lambda R$. $A / / R)\{R$. equiv $A R \wedge$ card $(A / / R)=k\}\{P$. partition-on $A$ $P \wedge \operatorname{card} P=k\}$
proof (rule bij-betw-byWitness[where $\left.\left.f^{\prime}=\lambda P .\{(x, y) . \exists X \in P . x \in X \wedge y \in X\}\right]\right)$ show $\forall R \in\{R$. equiv $A R \wedge \operatorname{card}(A / / R)=k\}$. $\{(x, y) . \exists X \in A / / R . x \in X \wedge$ $y \in X\}=R$
by (auto simp add: equiv-relation-of-partition-of)
show $\forall P \in\{P$. partition-on $A P \wedge$ card $P=k\}$. $A / /\{(x, y) . \exists X \in P . x \in X \wedge$ $y \in X\}=P$
by (auto simp add: partition-on-eq-quotient)
show $(\lambda R . A / / R)$ ' $\{R$. equiv $A R \wedge \operatorname{card}(A / / R)=k\} \subseteq\{P$. partition-on $A P \wedge \operatorname{card} P=k\}$
using partition-on-quotient by auto
show $(\lambda P .\{(x, y) . \exists X \in P . x \in X \wedge y \in X\})$ ' $\{P$. partition-on $A P \wedge$ card $P$ $=k\} \subseteq\{R$. equiv $A R \wedge \operatorname{card}(A / / R)=k\}$
using equiv-partition-on by (auto simp add: partition-on-eq-quotient)
qed

### 1.2 Finiteness of Equivalence Relations

lemma finite-equiv:
assumes finite $A$
shows finite $\{R$. equiv $A R\}$
proof -
have bij-betw $(\lambda R . A / / R)\{R$. equiv $A R\}\{P$. partition-on $A P\}$ by (rule bij-betw-partition-of)
from this show finite $\{R$. equiv $A R\}$
using 〈finite $A$ 〉 finitely-many-partition-on by (simp add: bij-betw-finite)
qed

### 1.3 Cardinality of Equivalence Relations

theorem card-equiv-rel-eq-card-partitions:
card $\{R$. equiv $A R\}=$ card $\{P$. partition-on $A P\}$
proof -
have bij-betw $(\lambda R$. $A / / R)\{R$. equiv $A R\}\{P$. partition-on $A P\}$
by (rule bij-betw-partition-of)
from this show card $\{R$. equiv $A R\}=$ card $\{P$. partition-on $A P\}$
by (rule bij-betw-same-card)
qed
corollary card-equiv-rel-eq-Bell:
assumes finite $A$
shows $\operatorname{card}\{R$. equiv $A R\}=\operatorname{Bell}(\operatorname{card} A)$
using assms Bell-altdef card-equiv-rel-eq-card-partitions by force

```
corollary card-equiv-rel-eq-sum-Stirling:
    assumes finite A
    shows card {R. equiv A R} = sum (Stirling (card A)) {..card A}
using assms card-equiv-rel-eq-Bell Bell-Stirling-eq by simp
theorem card-equiv-k-classes-eq-card-partitions-k-parts:
    card {R. equiv A R}\wedge\mathrm{ card (A// R)=k} = card {P.partition-on A P ^card
P=k}
proof -
    have bij-betw ( }\lambdaR.A//R){R. equiv A R\wedge card (A // R)=k} {P. partition-on
A P\wedge card P = k}
    by (rule bij-betw-partition-of-equiv-with-k-classes)
    from this show card {R. equiv }AR\wedge\mathrm{ card (A// R)=k}=card {P.partition-on
A P\wedge card P=k}
    by (rule bij-betw-same-card)
qed
corollary
    assumes finite A
    shows card {R. equiv A R}\wedge\operatorname{card (A // R)=k} = Stirling (card A)k
using card-equiv-k-classes-eq-card-partitions-k-parts
    card-partition-on[OF〈finite A〉] by metis
end
```


## 2 Cardinality of Partial Equivalence Relations

## theory Card－Partial－Equiv－Relations <br> imports <br> Card-Equiv-Relations <br> begin

## 2．1 Definition of Partial Equivalence Relation

definition partial－equiv ：：＇$a$ set $\Rightarrow\left({ }^{\prime} a \times{ }^{\prime} a\right)$ set $\Rightarrow$ bool where partial－equiv $A R=(R \subseteq A \times A \wedge$ sym $R \wedge$ trans $R)$
lemma partial－equivI： assumes $R \subseteq A \times A$ sym $R$ trans $R$ shows partial－equiv $A R$
using assms unfolding partial－equiv－def by auto
lemma partial－equiv－iff：
shows partial－equiv $A \quad$ p $\left(\exists A^{\prime} \subseteq A\right.$ ．equiv $\left.A^{\prime} R\right)$
proof
assume partial－equiv $A R$
from 〈partial－equiv $A R$ 〉 have $R$＂$A \subseteq A$
unfolding partial－equiv－def by blast

```
    moreover have equiv (R "A)R
    proof (rule equivI)
    from <partial-equiv A R` show sym R
        unfolding partial-equiv-def by blast
    from <partial-equiv A R` show trans }
        unfolding partial-equiv-def by blast
    show refl-on (R "A)R
    proof (rule refl-onI)
        show R\subseteqR" }A\timesR\mathrm{ " A
        proof
            fix p
            assume p\inR
            obtain }xy\mathrm{ where }p=(x,y)\mathrm{ by fastforce
            moreover have }x\inR"
                using }\langlep\inR\rangle\langlep=(x,y)\rangle\langlepartial-equiv A R
                    partial-equiv-def sym-def by fastforce
            moreover have }y\inR\mathrm{ " A
                using }\langlep\inR\rangle\langlep=(x,y)\rangle\langleR" "A\subseteqA\rangle\langlex\inR" "A\rangle by blas
            ultimately show }p\inR"A\timesR"A by aut
        qed
    next
        fix }
        assume }y\inR\mathrm{ " }
        from this obtain x where (x,y)\inR by auto
        from }\langle(x,y)\inR\rangle\mathrm{ have }(y,x)\in
            using <sym R` by (meson symE)
        from }\langle(x,y)\inR\rangle\langle(y,x)\inR\rangle\mathrm{ show }(y,y)\in
            using <trans R> by (meson transE)
        qed
    qed
    ultimately show }\exists\mp@subsup{A}{}{\prime}\subseteqA\mathrm{ . equiv }\mp@subsup{A}{}{\prime}R\mathrm{ by blast
next
    assume \exists}\mp@subsup{A}{}{\prime}\subseteqA\mathrm{ . equiv }\mp@subsup{A}{}{\prime}
    from this obtain }\mp@subsup{A}{}{\prime}\mathrm{ where }\mp@subsup{A}{}{\prime}\subseteqA\mathrm{ and equiv }\mp@subsup{A}{}{\prime}R\mathrm{ by blast
    show partial-equiv A R
    proof (rule partial-equivI)
        from <equiv A' R\rangle\langleA'\subseteqA\rangle show R\subseteqA\timesA
            using equiv-class-eq-iff by fastforce
        from \equiv A' R` show sym R
            using equivE by blast
    from <equiv A` R` show trans R
            using equivE by blast
    qed
qed
```


### 2.2 Construction of all Partial Equivalence Relations for a Given Set

definition all-partial-equivs-on :: 'a set $\Rightarrow\left(\left({ }^{\prime} a \times\right.\right.$ ' $\left.a\right)$ set $)$ set

```
where
    all-partial-equivs-on A =
        do {
            k\leftarrow{0..card A};
            A'\leftarrow{\mp@subsup{A}{}{\prime}.\mp@subsup{A}{}{\prime}\subseteqA\wedge card A'}=k}
            {R. equiv }\mp@subsup{A}{}{\prime}R
    }
lemma all-partial-equivs-on:
    assumes finite A
    shows all-partial-equivs-on }A={R\mathrm{ . partial-equiv A R}
proof
    show all-partial-equivs-on }A\subseteq{R.partial-equiv A R
    proof
        fix }
        assume R\in all-partial-equivs-on A
        from this obtain }\mp@subsup{A}{}{\prime}\mathrm{ where }\mp@subsup{A}{}{\prime}\subseteqA\mathrm{ and equiv }\mp@subsup{A}{}{\prime}
            unfolding all-partial-equivs-on-def by auto
        from this have partial-equiv A R
            using partial-equiv-iff by blast
            from this show }R\in{R\mathrm{ . partial-equiv A R} ..
    qed
next
    show {R. partial-equiv A R}\subseteq all-partial-equivs-on A
    proof
        fix }
        assume R \in{R. partial-equiv A R}
        from this obtain }\mp@subsup{A}{}{\prime}\mathrm{ where }\mp@subsup{A}{}{\prime}\subseteqA\mathrm{ and equiv }\mp@subsup{A}{}{\prime}
            using partial-equiv-iff by (metis mem-Collect-eq)
        moreover have card A'}\in{0..card A
            using <A' }\subseteqA\rangle\langlefinite A> by (simp add: card-mono
        ultimately show }R\in\mathrm{ all-partial-equivs-on }
            unfolding all-partial-equivs-on-def
            by (auto simp del: atLeastAtMost-iff)
    qed
qed
```


### 2.3 Injectivity of the Set Construction

lemma equiv-inject:
assumes equiv $A R$ equiv $B R$
shows $A=B$
proof -
from assms have $R \subseteq A \times A R \subseteq B \times B$ by (simp add: equiv-type) +
moreover from assms have $\forall x \in A .(x, x) \in R \forall x \in B .(x, x) \in R$
by (simp add: eq-equiv-class)+
ultimately show ?thesis
using mem-Sigma-iff subset-antisym subset-eq by blast
qed

```
lemma injectivity:
    assumes ( }\mp@subsup{A}{}{\prime}\subseteqA\wedge\operatorname{card}\mp@subsup{A}{}{\prime}=k)\wedge(\mp@subsup{A}{}{\prime\prime}\subseteqA\wedge\operatorname{card}\mp@subsup{A}{}{\prime\prime}=\mp@subsup{k}{}{\prime}
    assumes equiv A' R}\wedge\mathrm{ equiv }\mp@subsup{A}{}{\prime\prime}\mp@subsup{R}{}{\prime
    assumes R= R'
    shows A' = A'\prime }k=\mp@subsup{k}{}{\prime
proof -
    from }\langleR=\mp@subsup{R}{}{\prime}\rangle\operatorname{assms(2) show }\mp@subsup{A}{}{\prime}=\mp@subsup{A}{}{\prime\prime
        using equiv-inject by fast
    from this assms(1) show k= k' by simp
qed
```


## 2．4 Cardinality Theorem of Partial Equivalence Relations

```
theorem card-partial-equiv:
    assumes finite \(A\)
    shows card \(\{R\). partial-equiv \(A R\}=\operatorname{Bell}(\operatorname{card} A+1)\)
proof -
    let \(? \operatorname{expr}=d o\{\)
        \(k \leftarrow\{0\)..card \(A\} ;\)
        \(A^{\prime} \leftarrow\left\{A^{\prime} . A^{\prime} \subseteq A \wedge \operatorname{card} A^{\prime}=k\right\} ;\)
        \(\left\{R\right.\). equiv \(\left.A^{\prime} R\right\}\)
        \}
    have card \(\{R\). partial-equiv \(A R\}=\) card (all-partial-equivs-on \(A\) )
        using 〈finite \(A\) 〉 by (simp add: all-partial-equivs-on)
    also have card (all-partial-equivs-on \(A\) ) \(=\) card ? expr
            unfolding all-partial-equivs-on-def ..
    also have card ? expr \(=\left(\sum k=0 . . \operatorname{card} A .(\right.\) card A choose \(k) *\) Bell \(\left.k\right)\)
    proof -
        let ? \(S \gg\) ? \(c o m p=\) ? expr
    \{
        fix \(k\)
        assume \(k: k \in\{. . \operatorname{card} A\}\)
        let ? expr \(=\) ? comp \(k\)
        let ? \(S \gg\) ? comp = ? expr
        have finite ? \(S\) using 〈finite \(A\) by simp
        moreover \{
            fix \(A^{\prime}\)
            assume \(A^{\prime}: A^{\prime} \in\left\{A^{\prime} . A^{\prime} \subseteq A \wedge \operatorname{card} A^{\prime}=k\right\}\)
            from this have \(A^{\prime} \subseteq A\) and card \(A^{\prime}=k\) by auto
            let ? expr \(=\) ? comp \(A^{\prime}\)
            have finite \(A^{\prime}\)
                    using \(\langle\) finite \(A\rangle\left\langle A^{\prime} \subseteq A\right\rangle\) by (simp add: finite-subset)
            have card ? expr \(=\) Bell \(k\)
                    using \(\langle\) finite \(A\rangle\left\langle\right.\) finite \(\left.A^{\prime}\right\rangle\left\langle A^{\prime} \subseteq A\right\rangle\left\langle\right.\) card \(\left.A^{\prime}=k\right\rangle\)
                    by (simp add: card-equiv-rel-eq-Bell)
            moreover have finite ? expr
                using 〈finite \(A^{\prime}\) 〉 by (simp add: finite-equiv)
            ultimately have finite ? expr \(\wedge\) card ? expr \(=\) Bell \(k\) by blast
```

```
    \} note inner \(=\) this
    moreover have disjoint-family-on ?comp ?S
        by (injectivity-solver rule: injectivity(1))
    moreover have card ? \(S=\) card \(A\) choose \(k\)
        using \(\langle\) finite \(A\rangle\) by (simp add: \(n\)-subsets)
    ultimately have card ? expr \(=(\) card \(A\) choose \(k) *\) Bell \(k\) (is \(-=\) ?formula)
        by (subst card-bind-constant) auto
    moreover have finite ? expr
        using 〈finite ? S \(\downarrow\) inner by (auto intro!: finite-bind)
    ultimately have finite ?expr \(\wedge\) card ? expr \(=\) ?formula by blast
    \}
    moreover have finite ?S by simp
    moreover have disjoint-family-on ?comp ?S
    by (injectivity-solver rule: injectivity(2))
    ultimately show card ? expr \(=\left(\sum k=0 .\right.\). card \(A .(\operatorname{card} A\) choose \(k) *\) Bell \(\left.k\right)\)
    by (subst card-bind) auto
qed
also have \(\ldots=\left(\sum k \leq \operatorname{card} A .(\operatorname{card} A\right.\) choose \(k) *\) Bell \(\left.k\right)\)
    by (auto intro: sum.cong)
also have \(\ldots=\operatorname{Bell}(\operatorname{card} A+1)\)
    using Bell-recursive-eq by simp
finally show? thesis.
qed
end
```


## References

[1] L. Bulwahn. Cardinality of set partitions. Archive of Formal Proofs, Dec. 2015. http://www.isa-afp.org/entries/Card_Partitions.shtml, Formal proof development.
[2] L. Bulwahn. Spivey's generalized recurrence for bell numbers. Archive of Formal Proofs, May 2016. http://www.isa-afp.org/entries/Bell_ Numbers_Spivey.shtml, Formal proof development.
[3] N. J. A. Sloane. A000110: Bell or exponential numbers: number of ways to partition a set of n labeled elements. In The On-Line Encyclopedia of Integer Sequences. https://oeis.org/A000110.
[4] Wikipedia. Equivalence relation - wikipedia, the free encyclopedia, 2016. [Online; accessed 23-May-2016].

